

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Philbrick, et al.

Assignee: Alacritech, Inc.

Title: "TCP/IP Offload Device With Fast-Path TCP ACK Generating and Transmitting Mechanism"

Serial No.:

Filed: February 26, 2002

Examiner:

Group Art Unit:

Atty. Doc. No.: ALA-019

February 26, 2002

Box Patent Application
ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Sir:

Before examination on the merits, please amend the above-identified application as follows.

IN THE CLAIMS:

Claims 1-28 are replicated below with formatting:

1. A method of outputting a first TCP/IP packet and a second TCP/IP packet from a network interface device, the first TCP/IP packet and the second TCP/IP packet being output to a network, comprising:

- (a) storing first packet information on the network interface device;
- (b) pushing a first pointer to the first packet information onto a first transmit queue of the network interface device;
- (c) storing second packet information on the network interface device;
- (d) pushing a second pointer to the second packet information onto a second transmit queue of the network interface device; and
- (e) popping the second pointer off the second transmit queue and then popping the first pointer off the first transmit queue, the popped second pointer being used to

Applicants: Philbrick, et al.
Docket No.: ALA-019

locate the second packet information, the located second packet information then being output from the network interface device in the form of a second TCP/IP packet, the popped first pointer being used to locate the first packet information, the located first packet information being output from the network interface device in the form of a first TCP/IP packet such that the second TCP/IP packet is output from the network interface device and to the network before the first TCP/IP packet is output from the network interface device and to the network.

2. The method of Claim 1, wherein the first TCP/IP packet is a data packet, wherein the second TCP/IP packet is a control packet, and wherein the network interface device is coupled to a host computer by a parallel bus.
3. The method of Claim 1, wherein the first transmit queue contains pointers associated with a first set of packets, and wherein the second transmit queue contains pointers associated with a second set of packets, the second set of packets having transmission priority over the first set of packets.
4. The method of Claim 1, wherein the network interface device comprises a transmit sequencer, a memory, and MAC interface circuitry, the transmit sequencer causing the second packet information to be transferred from the memory to the MAC interface circuitry, the second TCP/IP packet being output from the network interface device through the MAC interface circuitry.
5. The method of Claim 1, wherein the first TCP/IP packet is associated with a first TCP/IP connection, and wherein the second TCP/IP packet is associated with a second TCP/IP connection, and wherein the pushing of (b) occurs before the pushing of (d).
6. The method of Claim 1, wherein the first packet information includes a header portion and a data payload portion.
7. The method of Claim 1, wherein the first pointer is part of a buffer descriptor.

Applicants: Philbrick, et al.
Docket No.: ALA-019

a processor that causes a first pointer to the first packet information to be pushed onto a first transmit queue before a second pointer to the second packet information is pushed onto a second transmit queue; and

a transmit mechanism that pops the second queue in preference to popping the first queue, the transmit mechanism popping the second pointer off the second queue and outputting the second packet information from the network interface device in the form of a second TCP/IP packet, the transmit mechanism popping the first pointer off the first queue and outputting the first packet information from the network interface device in the form of a first TCP/IP packet, the transmit mechanism popping the second pointer from the second queue before popping the first pointer off the first queue, the second TCP/IP packet being output from the network interface device before the first TCP/IP packet is output from the network interface device.

15. The TCP/IP offload network interface device of Claim 14, wherein the first TCP/IP packet is a data packet, and wherein the second TCP/IP packet is a control packet.

16. The TCP/IP offload network interface device of Claim 14, wherein the first TCP/IP packet is a data packet associated with a first TCP/IP connection, and wherein the second TCP/IP packet is a TCP ACK associated with a second TCP/IP connection.

17. The TCP/IP offload network interface device of Claim 16, wherein the TCP/IP offload network interface device is operatively coupled to a host computer, the host computer executing a protocol processing stack.

18. The TCP/IP offload network interface device of Claim 16, wherein the network interface device includes a second processor, the second processor executing a protocol processing stack, the second processor being part of the TCP/IP offload network interface device.

19. The TCP/IP offload network interface device of Claim 16, wherein a template header having TCP fields and IP fields is stored on the TCP/IP offload network interface device, and wherein the processor executes a transmit finite state machine, the transmit

Applicants: Philbrick, et al.
Docket No.: ALA-019

finite state machine filling in the TCP fields and the IP fields of the template header, the filled in template header forming at least a part of the second packet information.

20. The TCP/IP offload network interface device of Claim 19, wherein the transmit finite state machine does not include a TCP protocol processing layer and a discrete IP protocol processing layer, but rather the transmit finite state machine covers both TCP and IP protocol processing.

21. A method for outputting an acknowledge (ACK) from a protocol processing offload network interface device (PPONID), the PPONID being coupled to a network, the method comprising:

- receiving a first packet onto the PPONID from the network;
- slow-path processing the first packet such that a protocol processing stack performs substantial transport layer processing and substantial network layer processing on the first packet;
- receiving a second packet onto the PPONID from the network;
- fast-path processing the second packet on the PPONID such that the stack performs substantially no transport layer processing on the second packet and such that the stack performs substantially no network layer processing on the second packet;
- pushing a first pointer to first packet information onto a first transmit queue;
- in response to said receiving of the second packet pushing a second pointer to second packet information onto a second transmit queue;
- popping the second transmit queue to retrieve the second pointer, and using the second pointer to retrieve the second packet information, and outputting the second packet information from the PPONID in the form of the ACK; and
- popping the first transmit queue to retrieve the first pointer, and using the first pointer to retrieve the first packet information, and outputting the first packet information from the PPONID in the form of a third packet, the ACK being output from the PPONID before the third packet.

22. The method of Claim 21, wherein PPONID is coupled to a host computer, and wherein the second packet includes a data payload, the data payload being transferred

Applicants: Philbrick, et al.
Docket No.: ALA-019

from the PPONID and to the host, the ACK being output from the PPONID before any portion of the data payload is transferred to the host.

23. A method for outputting a TCP ACK from a network interface device, the network interface device being coupled to a network, the method comprising:

receiving a first TCP/IP packet onto the network interface device from the network;

slow-path processing the first TCP/IP packet such that a protocol processing stack performs substantial TCP layer processing and substantial IP layer processing on the first TCP/IP packet;

receiving a second TCP/IP packet onto the network interface device from the network;

fast-path processing the second TCP/IP packet on the network interface device such that the stack performs substantially no TCP layer processing on the second TCP/IP packet and such that the stack performs substantially no IP layer processing on the second TCP/IP packet;

pushing a first pointer to third packet information onto a first transmit queue;

in response to said receiving of the second TCP/IP packet pushing a second pointer to fourth packet information onto a second transmit queue;

popping the second transmit queue to retrieve the second pointer, and using the second pointer to retrieve the fourth packet information, and outputting the fourth packet information from the network interface device in the form of the TCP ACK; and

popping the first transmit queue to retrieve the first pointer, and using the first pointer to retrieve the third packet information, and outputting the third packet information from the network interface device in the form of a third TCP/IP packet, the TCP ACK being output from the network interface device before the third TCP/IP packet.

24. The method of Claim 23, wherein the protocol processing stack executes on a host computer, the network interface device being coupled to the host computer.

Applicants: Philbrick, et al.
Docket No.: ALA-019

25. The method of Claim 23, wherein the protocol processing stack executes on a processor, the processor being part of the network interface device.

26. The method of Claim 23, wherein the network interface device comprises a processor, the method further comprising the step of generating the fourth packet information, the step of generating the fourth packet information comprising:

accessing a template header stored on the network interface device, the template header having TCP fields and IP fields; and

executing a finite state machine on the processor, the finite state machine filling in the TCP fields and the IP fields and thereby forming the fourth packet information without passing the fourth packet information or the template header down to any lower protocol processing layer in any protocol stack.

27. The method of Claim 26, wherein the network interface device further comprises a transmit sequencer, the transmit sequencer causing the TCP ACK and the third TCP/IP packet to be output from the network interface device.

28. A method of generating an ACK on a TCP/IP offload device, comprising:

(a) step for using a template header to generate the ACK without sequentially processing the template header through a TCP protocol processing layer on the TCP/IP offload device and then processing the template header through an IP protocol processing layer on the TCP/IP offload device, the template header having TCP and IP fields, step (a) involving filling in the TCP and IP fields;

(b) pushing pointers to data packet information onto a first transmit queue, and then pushing a pointer to the filled-in template header onto a second transmit queue; and

(c) popping the second transmit queue in preference to popping the first transmit queue such that the second transmit queue has transmission priority over the first transmit queue, and outputting from the TCP/IP offload device the filled-in template header in the form of the ACK, the data packet information being output from the TCP/IP offload device in the form of data packets, the ACK being output from the

Applicants: Philbrick, et al.
Docket No.: ALA-019

TCP/IP offload device before any of the data packets are output from the TCP/IP offload device.

Please add new Claims 29-30 as follows.

- -29.(New) A TCP/IP offload device, the TCP/IP offload device being coupled to a device that executes a protocol stack, wherein the TCP/IP offload device accelerates TCP and IP protocol processing of an incoming TCP/IP packet such that the stack performs substantially no TCP protocol processing on the TCP/IP packet and such that the stack performs substantially no IP protocol processing on the TCP/IP packet, and wherein a data portion of the TCP/IP packet is transferred from the TCP/IP offload device to the device that executes the protocol stack, the TCP/IP offload device comprising:

a first transmit queue containing pointers associated with a first set of packets;
and

a second transmit queue containing pointers associated with a second set of packets, wherein the second set of packets has transmission priority over the first set of packets.

30.(New) The TCP/IP offload device of Claim 29, wherein the TCP/IP packet is received onto the TCP/IP offload device from a network, and wherein the TCP/IP offload device includes means for outputting the first set of packets and the second set of packets from the TCP/IP offload device and to the network.- -

Applicants: Philbrick, et al.
Docket No.: ALA-019

REMARKS

Consideration and allowance is respectfully requested.

Claims 1-28 as they appear in the application were printed out without formatting. These claims are therefore replicated here with formatting to assist the Examiner reviewing and considering the claims.

New Claims 29-30 are added.

After entry of this Preliminary Amendment, Claims 1-30 are pending. If the Examiner would like to discuss any aspect of this application, the Examiner is requested to contact the undersigned at (925) 485-9923.

I hereby certify that this correspondence is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR §1.10 on the date indicated below and is addressed to:

Box Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

By: Lester Wallace

Typed Name: T. Lester Wallace
Express Mail Label No.: EL928549021US
Date of Deposit: February 26, 2002.

Lester Wallace Feb. 26, 2002
Signature Date of Signature

Respectfully submitted,

Lester Wallace

T. Lester Wallace
Attorney for Applicants
Reg. No. 34,748

